Internet of Things

Internet of Things and NodeMCU

IoT Team, BFCAI



IoT Applications















← STYLE CHECK RESULTS













Internet of Things

 The Internet of Things (IoT) represents the network of physical objects "Things" that are integrated with sensors, software and other technologies for the purpose of exchanging data with other devices on the Internet.



- A sensor is a device that detects some type of input from the physical environment.
- The input can be light, heat, motion, pressure or any number of other environmental phenomena.



Sensors









Temperature and Humidity

PIR Motion Detection

Microphone Sound Detection

Gas/Smoke Sensor









Photoresistor CdS Sensor Ultrasonic Sensor IR Obstacle Avoidance Heart Rate Sensor (ECG)

- Sensors turn a physical input into an electrical output, while actuators do the opposite.
- Actuators take electrical signals from control modules and turn them into physical outputs.







Arduino Uno

Arduino Mega

Processing – Arduino R4 Wi-Fi



Arduino UNO R4 WIFI (China Version)

⊘ In Stock

EGP1,400.00



Add to wishlist

Categories: Arduino, Arduino Boards



Processing – Raspberry Pi



Raspberry Pi

Processing – ESP8266, ESP32 & ESP32-CAM



NodeMCU ESP8266

ESP32

ESP32-CAM

NodeMCU ESP8266 or ESP32?

- The ESP8266 and the ESP32 are both low-cost microchips with Wi-Fi and microcontroller capabilities, making them well-suited for IoT applications.
- Performance: The ESP32 has a more powerful processor and more RAM than the ESP8266.
- **Connectivity**: The ESP32 supports Bluetooth in addition to Wi-Fi, which makes it better for applications that require both wireless technologies.
- Power consumption: The ESP32 has a lower power consumption than the ESP8266.
- Price: The ESP8266 (270 EGP) is cheaper than the ESP32 (430 EGP), making it a more budget-friendly option for projects.

Different Types of Wi-Fi Modules



ESP-11





ESP-13





ESP-12 Versions



ESP-12 Wi-Fi Module

- ESP8266 is a cost-effective, easy-to-operate, compact-sized and lowpowered Wi-Fi module.
- It supports both TCP/IP and Serial Protocols.
- It's normally used in IoT-based embedded projects and is considered the most widely used Wi-Fi module because of its low cost and small size.



NodeMCU ESP8266: Versions

ESP8266 CP2102



ESP8266 CH340 V3



NodeMCU ESP8266: Versions



ESP8266 CP2102

ESP8266 CH340 V3

NodeMCU ESP8266 V3

- NodeMCU is a low-cost open-source IoT platform based on the ESP8266
 Wi-Fi system on a chip.
- NodeMCU Version 3 runs on the ESP-12E (ESP8266MOD) module, and it is easy-to-use development board equipped with analog and digital pins, a USB-to-serial adapter based on CH340G module, and a micro-USB.



- Open-source
- Low Cost
- Arduino-like Hardware
- ESP8266 with Built-in Wi-Fi
- Programmable
- GPIO Pins
- PWM
- Status LED
- MicroUSB Port
- USB-to-UART Converter

NodeMCU ESP8266 V3: Pinout



NodeMCU ESP8266 V3: Pinout



Microcontroller	L106 32-bit RISC
Clock Speed	80 MHz
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins (ADC)	1
ADC Range	0-3.3V (10-bit)
PWM Resolution	0-1023 (10-bit)
Flash Memory	4 MB
SRAM	64 KB
USB-to-Serial	CH340G

Model	ESP8266-12E
Wireless Standard	802.11 b/g/n
Frequency Range	2.4 GHz
Wi-Fi Mode	Station / AP / AP+Station
Stack	Integrated TCP/IP
Data Interface	UART / 12C / HSPI / 12S
Encryption	WEP / TKIP / AES
Built-In WiFi	Yes
Built-In Bluetooth	Νο
USB Connector	Micro USB

NodeMCU ESP8266 V3: Digital and Analog Pins

PIN	GPIO	Туре
D0	GPIO16	Digital
D1	GPIO5	Digital
D2	GPIO4	Digital
D3	GPIO0	Digital
D4	GPIO2	Digital
D5	GPIO14	Digital
D6	GPIO12	Digital
D7	GPIO13	Digital
D8	GPIO15	Digital
A0	ADC0	Analog



NodeMCU ESP8266 V3: GPIOs Safe to Use

PIN	GPIO	Why Not Safe?
D0	GPIO16	HIGH at boot Used to wake up from deep sleep
D1	GPIO5	-
D2	GPIO4	-
D3	GPIO0	Connected to FLASH button Boot fails if pulled LOW
D4	GPIO2	HIGH at boot Boot fails if pulled LOW
D5	GPIO14	-
D6	GPIO12	-
D7	GPIO13	-
D8	GPIO15	Required for boot Boot fails if pulled HIGH



NodeMCU ESP8266 V3: GPIOs Safe to Use

PIN	GPIO	Why Not Safe?		
RX	GPIO3	Used for flashing and debugging		
тх	GPIO1	Used for flashing and debugging		
CLK	GPIO6	Connected to Flash memory		
SDO	GPIO7	Connected to Flash memory		
CMD	GPIO11	Connected to Flash memory		
SD1	GPIO8	Connected to Flash memory		
SD2	GPIO9	Connected to Flash memory		
SD3	GPIO10	Connected to Flash memory		
A0	ADC0	Analog input pin Cannot be configured as output		



NodeMCU ESP8266 V3: PWM Pins

- All of the ESP8266's GPIO pins, from GPIO0 to GPIO15, can be programmed to generate pulse width modulated (PWM) outputs.
- On the ESP8266, the PWM signal has a 10-bit resolution.

```
analogWrite(pin, 0);
analogWrite(pin, 1023);
```





NodeMCU ESP8266 V3: Power Pins

PIN	Туре
G	Ground Pin
3.3V	3.3V Power Pin
VIN	Can be Used to Directly Power the NodeMCU/ESP8266
VU	Voltage Supplied by the USB



Arduino IDE

• The Arduino IDE enables you to write and edit code and convert this code into instructions that NodeMCU hardware understands.



• First, open the Arduino IDE and go to File \rightarrow Preferences.

🥯 ske	etch_nov08	a Arduino 1.8.1	8			×
File Ed	lit Sketch	Tools Help				
Ne	ew	Ctrl+N				9
Op	pen	Ctrl+O				
Op	pen Recent	;				
Ske	etchbook	>				^
Exa	amples	>				
Clo	ose	Ctrl+W	etup code here, to run once:			
Sa	ive	Ctrl+S				
Sa	ave As	Ctrl+Shift+S				
Pa	ige Setup	Ctrl+Shift+P				
Pri	int	Ctrl+P				
Pre	eferences	Ctrl+Comma				
Qu	uit	Ctrl+Q	ain code here, to run repeatedly:			
			۵ 			
}						~
1				Arduin	o Uno on (сом7

• On the Additional Boards Manager URLs, paste the following URL.

https://arduino.esp8266.com/stable/package_esp8266com_index.json

🥯 sketch_nov08a A	rduino 1.8.18			— [x c
File Edit Sketch Tool	s Help				
	Preferences		×		.
	Settings Network				_
sketch_nov08a	Sketchbook location:				M
void setur	C:\Users\Ghamry\Documents	Arduino	Browse		
// put y	Editor language:	System Default			
	Editor font size:	20			
}	Interface scale:	Automatic 100 % (requires restart of Arduino)			
,	Theme:	Default theme $$			
	Show verbose output during:	compilation upload			
void loop	Compiler warnings:	None V			
// put	Display line numbers	Enable Code Folding			
	Verify code after upload	Use external editor			
1	Check for updates on star	tup Save when verifying or uploading			
}	Use accessibility features				•
	Additional Boards Manager UF	Ls: https://arduino.esp8266.com/stable/package_esp8266com_index.json			
	More preferences can be edite	ed directly in the file			
	C:\Users\Ghamry\AppData\Lo	cal\Arduino15\preferences.txt			
	(edit only when Arduino is not	running)			
		OK	Cancel		

• Now go to Tools \rightarrow Board \rightarrow Boards Manager.

🥯 sketch_nov08a Arduino 1.8.18						
File Edit Sketch T	ools Help					
	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T				
void se	Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L	ru	in once:		
	WiFi101 / WiFiNINA Firmware Updater		-		1	
}	Board: "Arduino Uno" Port Get Board Info	>	•	Boards Manager Arduino Yún Arduino Uno		
void 100 // pu [.]	Programmer: "AVRISP mkll" Burn Bootloader	>		Arduino Uno Mini Arduino Duemilanove or Diecimila Arduino Nano		
}				Arduino Mega or Mega 2560 Arduino Mega ADK		

• Search ESP8266, and install esp8266 by ESP8266 community.

∞ Boards Manager	×
Type All V ESP8266	
esp8266 by ESP8266 Community Boards included in this package: Generic ESP8266 Module, Generic ESP8285 Module, Lifely Agrumino Lemon v4, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, WiFi Kit 8, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SparkFun Blynk Board, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 ESP-WROOM-02, LOLIN(WEMOS) D1 mini (clone), LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, LOLIN(WeMos) D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core, Schirmilabs Eduino WiFi, ITEAD Sonoff, DOIT ESP-Mx DevKit (ESP8285). Online Help More Info	^
	L
	~
Close	e

Installing NodeMCU CH340 Driver

• Now, connect your NodeMCU ESP8266, and install the CH340 Driver.



- Ports (COM & LPT)
 - ECP Printer Port (LPT1)
 - Intel(R) Active Management Technology SOL (COM3)
 - USB-SERIAL CH340 (COM5)

Installing NodeMCU CP2102 Driver

• Install the CP2102 Driver.



v6-7-6-driver-release-notes.txt

Sketches

• A sketch is the name that Arduino uses for a program.

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Sketches

- There are two special functions that are a part of every Arduino sketch: setup() and loop().
- The setup() is called once, when the sketch starts.
- It's a good place to do setup tasks like setting pin modes.
- The loop() function is called over and over and is heart of most sketches.
- You need to include both functions in your sketch, even if you don't need them for anything.

Your First NodeMCU Project: Blinking the On-board LED

• Turn the on-board LED on and off every second.



```
#define LED_PIN 2 // D4 (GPI02)
```

```
// The setup function runs once when you press reset or power the board
void setup() {
   pinMode(LED_PIN, OUTPUT); // Initialize the pin D4 as an output
}
```

```
// The loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off
  delay(1000); // Wait for a second
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on
  delay(1000); // Wait for a second
```

```
#define LED_PIN D4 // D4 (GPI02)
```

```
// The setup function runs once when you press reset or power the board
void setup() {
   pinMode(LED_PIN, OUTPUT); // Initialize the pin D4 as an output
}
```

```
// The loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off
  delay(1000); // Wait for a second
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on
  delay(1000); // Wait for a second
```

Your First NodeMCU Project: Arduino AVR Boards

- Now go to Tools \rightarrow Board \rightarrow ESP8266 Boards.
- Select NodeMCU 1.0 (ESP-12E Module).

🥯 Blir	nk Arduir	no 1.8.18					- [×
File Edi	t Sketch	Tools Help							
		Auto Format Archive Sketch	Ctrl+T			▲ Arduino DOIT ESP-Mx DevKit (ESP	8285)	3).
Blink #de:	§ fine	Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L	// D4 (GE	21	Digistump Oak ESPDuino (ESP-13 Module ESPectro Core ESPino (ESP-12 Module)	e)		^
11	The :	WiFi101 / WiFiNINA Firmware Update	r	en you press	5	ESPresso Lite 1.0		oar	.c
voi p	d se inMo	Board: "Arduino Uno" Port: "COM5" Get Board Info	د د	Boards Manager Arduino AVR Boards ESP8266 Boards (3.1.2	> () >	ESPresso Lite 2.0 ITEAD Sonoff Invent One		1 00	ıt
}		Programmer: "AVRISP mkll" Burn Bootloader	>	• •		LOLIN(WEMOS) D1 ESP-W LOLIN(WEMOS) D1 R2 & I LOLIN(WEMOS) D1 mini (mini clone)		> ~
						LOLIN(WEMOS) D1 mini L LOLIN(WEMOS) D1 mini F LOLIN(WeMos) D1 R1	ite Pro		^
Leav Haro	ving. d res	••• setting via RTS pin.				Lifely Agrumino Lemon v4 NodeMCU 0.9 (ESP-12 Mc NodeMCU 1.0 (ESP-12E M Olimey MOD-WIEL-ESP82)	f odule) lodule)		~
<						Phoenix 1.0	00(020) -		>

Your First NodeMCU Project: Port

• Go to Tools \rightarrow Port, and select the port of your NodeMCU board.

🥺 02.2Blink_BCM	I Arduino 1.8.18	
File Edit Sketch	Tools Help	
	Auto Format	Ctrl+T
	Archive Sketch	
02.2Blink_BCN	Fix Encoding & Reload	
#define	Manage Libraries	Ctrl+Shift+I Ctrl+Shift+I PTO2)
" "	Serial Monitor	Ctrl+Shift+I
// The :	WiFi101 / WiFiNINA Firmware Updater	reset or power the board
void se	Board: "NodeMCU 1.0 (ESP-12E Module)"	>
ninMo	Builtin Led: "2"	lize the pip D4 as an output
P TIMIO	Upload Speed: "115200"	> The the pin bi as an eacpac
}	CPU Frequency: "80 MHz"	>
	Flash Size: "4MB (FS:2MB OTA:~1019KB)"	>
// The	Debug port: "Disabled"	forever
woid lo	Debug Level: "None"	
VOIU IO	VTables: "Flash"	
digita	C++ Exceptions; "Disabled (new aborts on oom)"	he LED off
delay	Stack Protection: "Disabled"	or a second
digita	Erase Flash: "Only Sketch"	he LED on
digita	SSL Support: "All SSL ciphers (most compatible)"	
delay	MMU: "32KB cache + 32KB IRAM (balanced)"	, or a second
}	Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM	
	Port: "COM5"	> Serial ports
	Get Board Info	COM3
	Programmer	X COMS

Your First NodeMCU Project: Upload a Sketch

• Click the Upload button to program your NodeMCU with the sketch.

∞ 02.2Blink_BCM Arduino 1.8.18		- 🗆	×
File Edit Sketch Tools Help			
			<mark>.</mark>
02.2Blink_BCM			
#define LED_PIN D4	// D4 (GPIO2)		^
<pre>// The setup function runs once when void setup() {</pre>	you press reset or power	the b	oard
<pre>pinMode(LED_PIN, OUTPUT); }</pre>	// Initialize the pin D4	as an	out
<pre>// The loop function rung over and o </pre>	vor again forovor		>
Done Saving.			
			^
Leaving			
Hard resetting via RTS pin			~
<			>
compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5			

Analog vs. Digital Signals





Reading Analog Voltage

- The ESP8266-12E module has one ADC pin.
- The ADC pin has a 10-bit resolution, which means you'll get values between 0 and 1023.

value = analogRead(A0);



Photoresistor (Light Sensor)

• The photoresistor is a lightsensitive, variable resistor.



Photoresistor (Light Sensor)

Photoresistors are perfect for making light-controlled switches.



Photoresistor: Circuit



Photoresistor: Components

Now, we need the following components

- NodeMCU
- LED
- Photoresistor
- 330Ω Resistor
- 10KΩ Resistor
- Jumpers
- Breadboard

1. Connect breadboard power (+) and ground (-) rails to NodeMCU

3.3V and GND, respectively.



2. Drag a photoresistor to your breadboard, so its legs plug into two different rows.



3. Create a wire connecting one photoresistor leg to power.



4. Drag a $10K\Omega$ resistor to connect other photoresistor leg to the ground.



5. Connect the photoresistor leg that is connected with the ground to the NodeMCU A0 pin.



6. Plug the LED into two different breadboard rows.



7. The cathode (shorter leg) connects to one leg of a resistor of 330Ω , and the other resistor leg to the ground.



8. Wire up the LED anode (longer leg) to NodeMCU pin D1.



Photoresistor: Code

```
int photoresistor = 0;
int threshold = 750;
#define LED PIN D1
```

```
void setup()
  Serial.begin(9600);
```

```
pinMode(LED PIN, OUTPUT);
```

```
void loop()
```

```
Serial.println(photoresistor);
```

```
if (photoresistor < threshold)</pre>
  digitalWrite(LED PIN, HIGH);
else
```

digitalWrite(LED PIN, LOW);

delay(100);

```
// Define a that holds a the photoresistor reading
// Define a threshold variable
```

// Define LED PIN D1 (GPI05)

// Start serial monitor // Set LED PIN as an output pin

- photoresistor = analogRead(A0); // Read the brightness of the light // Print the value of photoresistor on serial monitor
 - // If the photoresistor value is below the threshold // Turn on the LED // Else // Turn off the LED

```
// Short delay
```

Assignment: Traffic Light



References and Tutorials

- IoT NodeMCU (ESP8266) Mohamed Yousef
- ESP8266 Pinout
- A Complete Guide on ESP8266 WiFi Based Microcontroller
- NodeMCU ESP8266 Detailed Review
- NodeMCU ESP8266 Arduino Store
- Introduction to NodeMCU V3
- ESP8266 Pinout Reference
- <u>Get Started with Arduino IDE and ESP8266-NodeMCU</u>
- How to Install CH340 Drivers
- NodeMCU PWM with Arduino IDE
- ESP8266 ADC Read Analog Values